Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona

UNIVERSITAT POLITÈCNICA DE CATALUNYA

SIGNAL GRAPHS AND NETWORKS

by

David Grau Marina

A Degree's Thesis submitted to the Faculty of the
Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona,
Universitat Politècnica de Catalunya

Advisor

Prof. Gregori Vázquez Grau

In partial fulfillment of the requirements for the Degree in
Telecommunication Systems Engineering

Barcelona, June 2017

Abstract

Due to the technologies improvement, the volume of data generated in any discipline has become massive. All this raw knowledge needs a post-processing in order to discern and identify the embedded information. Hence, it is necessary to create a new investigation path in discrete signal processing in order to mitigate this problem. This degree's thesis proposes a signal study with classical discrete processing techniques on graphs (DSP_G).

First, some basic concepts of graphs theory are defined and discrete signal processing tools are provided, such as filtering or frequency domain transformation, oriented to this methodology. Then, two interesting applications are developed: signal compression and linear prediction. It is demonstrated that graphs techniques are valid in order to treat very large signals, improving the results of classical methods.

Finally, it is addressed the issue of identifying the ideal structure of a graph from the information of its nodes. This problem stands for a novel research topic. Lot of literature aim its own methodologies to achieve the solution. In this work it is identified the optimal structure by minimizing the number of connections on the graph in order to avoid redundancy and denoising introduced by incorrelated nodes.

Resumen

Con el avance de las tecnologías, el volumen de datos generados en cualquier disciplina se ha masificado. Todo este conocimiento bruto necesita de un post-procesado con tal de discernir e identificar la información contenida. Es por ello que es necesario abrir una nueva ventana en el procesado de señales discretas con tal de tratar este problema. En esta tesis se propone el estudio de señales con técnicas del clásico procesado discreto orientado a grafos (DSP_G).

Primeramente, se definen los conceptos básicos en teoría de grafos, y se adquieren conocimientos en técnicas de procesado, tales como filtrado o transformaciones en el dominio frecuencial, orientado a esta metodología. Seguidamente, se desarrollan dos aplicaciones de interés como son la compresión de señales y la predicción lineal. Se demuestra que las técnicas empleadas mediante grafos son muy válidas para tratar señales de mucho volumen, mejorando los métodos clásicos.

Finalmente, se aborda el problema de identificar la estructura ideal de un grafo teniendo en cuenta la información anexada a sus nodos. Esta cuestión es un tema de investigación muy novedoso. Mucha literatura propone sus métodos para lograr este fin. En este trabajo se desarrolla el método de identificar una estructura óptima minimizando el número de conexiones del grafo con tal de reducir la redundancia y la distorsión introducida por posibles nodos incorrelados.

Resum

Amb l'avanç de les tecnologies, el volum de dades generat en qualsevol disciplina s'ha massificat. Tot aquest coneixement brut requereix d'un post-processat per tal de poder discernir i identificar la informació. És per això que es necessari obrir una nova finestra en el processat de senyals discretes per tractar aquest problema. En aquesta tesis es proposa l'estudi de senyals amb tècniques del processat discret orientat a grafs (DSP_G).

Primerament, es defineixen els conceptes bàsics en teoria de grafs, i s'adquireixen coneixements en tècniques de processat, tals com el filtrat o transformacions en el domini freqüencial, orientat a aquesta metodologia. Seguidament, es desenvolupen dues aplicacions d'interès, com son la compressió de senyals i la predicció lineal. Es demostra que les tècniques utilitzades mitjançant grafs son molt vàlides per tractar senyals de molt volum, millorant els mètodes clàssics.

Finalment, s'aborda el problema d'identificar l'estructura ideal d'un graf tenint en compte la informació annexada als seus nodes. Aquesta qüestió és un tema d'investigació molt innovador. Molta literatura proposa els seus mètodes per assolir aquest fi. En aquest treball es desenvolupa el mètode d'identificar una estructura òptima minimitzant el nombre de connexions del graf per tal de reduir la redundància i la distorsió introduïda per possibles nodes incorrelats.

Agraïments

Aquest treball culmina una etapa molt important de la meva vida. És el punt i final a quatre anys d'intens treball i dedicació, amb moments bons i d'altres no tant. Gràcies als meus estudis, he pogut descobrir tot un món ple de coneixement i innovació. A més a més, la tendència actual ens permet descobrir nous camins a cada moment.

En primer lloc, voldria expressar les meves gràcies al meu cercle més proper. La meva família i la meva parella m'han recolzat en cada decisió d'aquest llarg i sinuós camí. Ells són la raó de que jo sigui qui sóc avui dia. Aquest treball va dedicat a ells.

Voldria expressar també les meves sinceres gràcies al professor Gregori Vázquez. Durant les seves classes em vaig adonar que era un pou de coneixement i experiència, la qual cosa m'ha permès aprendre d'ell i amb ell. També voldria expressar el meu agraïment a en Jordi Borràs, per haver assistit a les nostres reunions i orientar-me i donar-me un cop de mà quan ho he necessitat.

Finalment, i no per això menys important, voldria agrair als meus amics de la facultat. Tot el que he après ho he compartit amb ells. També els moments de diversió. El camí continua.

List of Figures

Notation

$\mathbf{x}$	A column vector.
$\mathbf{X}$	A matrix.
$\mathbf{I}$	The identity matrix.
$\mathbf{0}$	All-zeros matrix or vector.
$\mathbf{1}$	All-ones column vector.
$\mathcal{X}$	A set of subspace.
$\text{rank}(\mathbf{X})$	The rank of $\mathbf{X}$.
$\text{diag}(\mathbf{X})$	A column vector containing the diagonal entries of $\mathbf{X}$.
$\text{diag}(\mathbf{x})$	A matrix whose diagonal contains the elements of $\mathbf{x}$.
$\mathbf{X}^{\#}$	The Moore-Penrose pseudoinverse of $\mathbf{X}$.
$(\cdot)^*$	The conjugate operator.
$(\cdot)^T$	The transpose operator.
$(\cdot)^H$	The hermitian (transpose and conjugate) operator.
$\odot$	The Hadamard product.
$\otimes$	The Kronecker product.
$\diamond$	Khatri-Rao (column-wise Kronecker) product.
$\mathbb{E}$	The mathematical expectation operator.
$\|\|\mathbf{x}\|\|$	The Euclidian norm of $\mathbf{x}$.
$\|\|\mathbf{X}\|\|_1$	The ℓ_1 norm of $\mathbf{X}$.
$\|\|\mathbf{X}\|\|_2$	The spectral norm of $\mathbf{X}$.
$\mathbb{R}$	The field of real numbers.
$\mathbb{C}$	The field of complex numbers.

Contents

1 | Introduction

1.1 Project Outline

Discrete signal processing on graphs is a novel field requiring an intensive and solid study. The scope of this project is to set up a solid background in graph signals. Moreover, we address the problem of identifying the graph structure from the signals on its nodes.

1.1.1 Requirements and Assumptions

The development of the project is described in the following lines. In order to make an accurate study of graph signals and to identify some of their applications, the work has been basically broken down into four main work-packages:

- Problem statement: in this starting point, it was required to get some knowledge and tools to work with graph signals, in order to develop some techniques and applications related to this field.

- Data acquisition and manipulation: in order to simulate and developing some applications on graphs, it is required to interact with large data sets. The chosen data was provided by the United States National Climatic Data Center (USNCDC) [1], which holds a huge database from lots of climatological measurements all around the American territory.

- Study of the algorithms and implementation: it is required to identify, study and implement some algorithms and techniques on graphs.

- Data analysis and results validation: it is mandatory to verify simulations and algorithms results. Moreover, simulations back up theoretical study and conclude the validity of the theories.

Further information about time planning and realization of the different work-packages is shown in appendix A, including the Gantt diagram.

1.2 Organization

This degree's thesis is organized into five chapters. The fist one, chapter 1, includes the introduction of the work, where the project scopes and requirements are analyzed. This chapter presents the structure of the project, and introduces and reference the starting point of the project. Additionally, useful information that facilitate the understanding of the projects is included before chapter 1, such as list of figures or notation.

In chapter 2 we introduce graph signals and some fundamental signal processing techniques applied to graphs, such as filtering and frequency analysis. In chapter 3, we use the tools introduced in the previous section to develop two interesting applications. The first one consists in exploting the frequency analysis to perform signal compression, whereas the second one deals with an important application of graph signals filtering. In chapter 4 we approach the problem of identifying the optimal structure of the graph. We determine some requirements and constraints on the graph, in order to make the problem affordable.

Finally, chapter 5 includes the conclusions and future tasks in order to make a complete analysis of the topics introduced in this thesis.

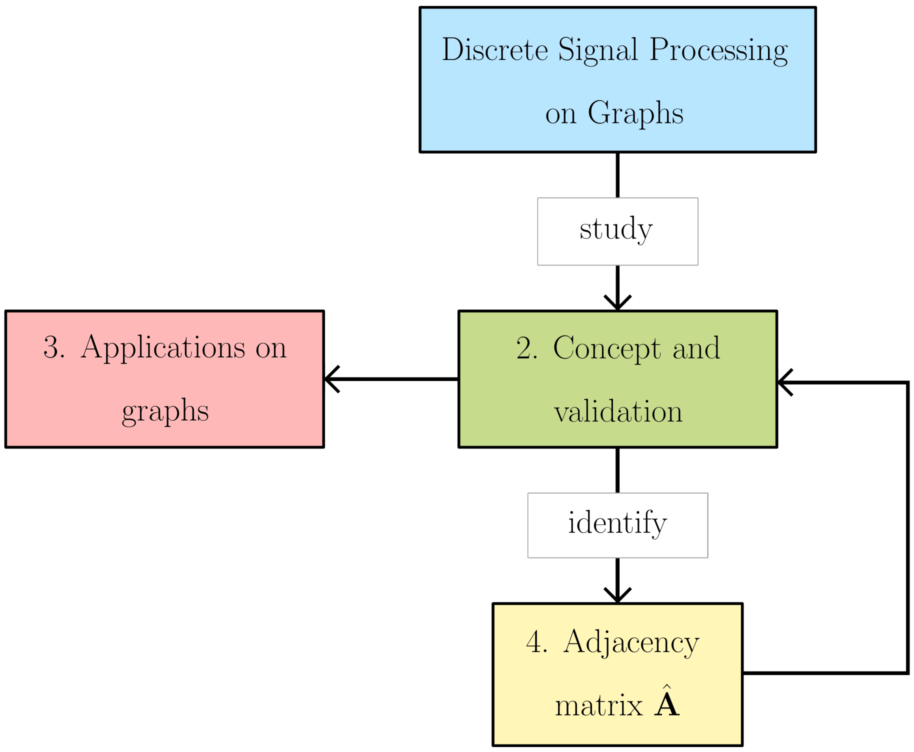
Discrete Signal Processing on Graphs
study
3. Applications on graphs
2. Concept and validation
identify
4. Adjacency matrix $\hat{\mathbf{A}}$

Figure 1.1: Degree's thesis structure.

1.3 State of the art

In the recent times, massive data sets are generated constantly in any discipline. From industrial applications to social networks. The large volume of raw data generated require accurate and optimal techniques to process the information, but their volume and complex structure limit the tools applied in small data sets. Analysis and processing this very large data sets, known as big data, stands for a big challenge. New discrete signal processing (DSP) techniques have to be applied [2].

In this thesis it is reviewed data analysis based on the classical discrete signal processing generalized to graphs (DSP_G). This modern discipline, which is in constant development, poses new investigation fields to develop. In graphical models, data is indexed by the nodes of the graph. Hence, it can be easily seen that resulting signal remove any dependencies in the original signal domain, such as time, frequency or space [6]. The goal of DSP_G is to develop a solid framework and tools for any data sets [3]. Graphs models can be directed or undirected. In this work we define tools oriented to undirected graphs. Some of the definitions, such as Fourier transform on graphs, is not applicable to directed graphs, which often happen in real-world in non-stationary problems.

From the literature, we discuss how fundamental signal processing methodologies, such as frequency analysis [7]–[8] or filtering [5] can be implemented efficiently for large data sets. DSP_G exploit these techniques from the structure of the graph. The structure determines how data elements are related, that is by dependency, physical proximity, similarity or any other property.

This main concept, the study of the structure of the graph, opens a new field of study. Which is the optimal structure for a graph? This too generic and non-trivial problem leads us to an intensive research topic. Furthermore, this simple question hide some important characteristics and behaviors of the graph, such as constraints in the connections between the nodes (or data elements). Again, some literature tries to explain and develop some algorithms in order to estimate the optimal structure of a graph, no matter the nature of the original data set [9].

2 | Graph signals

This chapter is devoted to review main concepts of DSP_G. First, we define graph signals and its notation. We can consider important operations on graph signals, such as graph shifting and graph filtering. Finally, we extend Fourier transform from DSP to graphs.

In order to illustrate all the definitions and applications, in this thesis we work with a large data set obtained from the United States National Climatic Data Center (USNCDC) [1].

Figure 2.1 shows an example of a graph signal. The values at nodes refers to an average over last thirty years of the maximum temperature registered by weather sensors. Edges represent the connection between nodes of the graph, defining its structure.

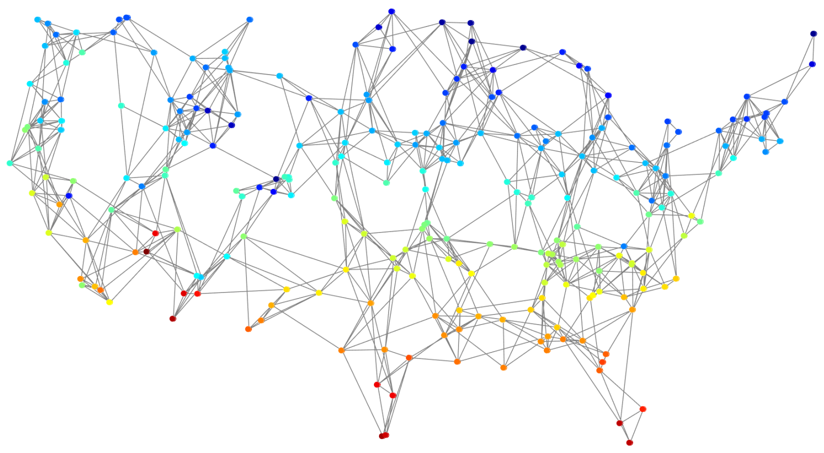

Figure 2.1: Maximum average temperature measurements (°C) across the United States.

2.1 Graph signals

In graphical models, data is indexed by the nodes of the graph. The structure determines how data elements are related, that is by dependency, physical proximity, similarity or any other property.

From the large data set previously introduced, we can construct a representation graph $G = (\mathcal{V}, \mathbf{A})$, where $\mathcal{V} = \{v_0, ..., v_{N-1}\}$ is the set of N nodes and $\mathbf{A}$ is the weighted adjacency matrix of the graph [3]. From figure 2.1, each node v_n corresponds to a climatological sensor, making temperature measurements.

Given this graph, the set of measurements at nodes forms a graph signal, defined as

$$\mathbf{s} = [\ s_0 \quad s_1 \quad \dots \quad s_{N-1}\ ]^T\,, \tag{2.1}$$

where each value s_n is related to the node v_n. In general, it is appropriate to consider $s_n \in \mathbb{C}$, defining the subspace $\mathcal{S}$ of graph signals. A graph is also represented by the adjacency matrix $\mathbf{A}$. This matrix defines the relations between the nodes. Hence, it defines the structure of the graph. In this chapter, the procedure used to build the graph is based on the similarity of the nodes based on the distances between them, so $a_{m,n} \in \mathbb{R}$ [2]-[3]. Each node is connected to K nearest sensors, known as neighbors. The set of indices of nodes connected to v_n is called neighborhood of v_n and it is denoted as

$$\mathcal{N}_n = \{m \mid a_{n,m} \neq 0\}\,. \tag{2.2}$$

In this chapter, we consider nodes connected with undirected edges weighted depending on the distance $d_{n,m}$ between the nth and the mth sensor [2], where $m \in \mathcal{N}_n$, so

$$a_{m,n} = \frac{e^{-d_{n,m}^2}}{\sqrt{\sum_{k\in\mathcal{N}_n} e^{-d_{n,k}^2} \sum_{l\in\mathcal{N}_m} e^{-d_{m,l}^2}}}\,. \tag{2.3}$$

The adjacency matrix $\mathbf{A}$ contains information of the neighborhood connections, as well as the value of this connections. Hence, the adjacency matrix can be decomposed into two matrices, that is

$$\mathbf{A} = \mathbf{B} \odot \mathbf{W}\,, \tag{2.4}$$

where $\mathbf{B}$ is the edges matrix containing only the connections between nodes, so $b_{m,n} \in \{0,1\}$. Matrix $\mathbf{W}$ contains the weighting information of the edges, so $w_{m,n} \in \mathbb{C}$. Note that matrices are related by the Hadamard product [11]. For two general $M \times N$ matrices $\mathbf{X}$ and $\mathbf{Y}$, the Hadamard product $(\mathbf{X} \odot \mathbf{Y})$ is a matrix of the same dimensions as the operands, whose entries are given by

$$(\mathbf{X} \odot \mathbf{Y})_{m,n} = (x_{m,n})(y_{m,n}) \qquad 0 \leqslant m < M\ ,\ 0 \leqslant n < N\,. \tag{2.5}$$

2.2 Graph shift

The most simple operation to be done with the graph signal is called graph shift. It consists in filtering the input graph signal $\mathbf{s}$ with the adjacency matrix $\mathbf{A}$ related to the graph, so

$$\widetilde{\mathbf{s}} = [\ \widetilde{s}_0 \quad \widetilde{s}_1 \quad \dots \quad \widetilde{s}_{N-1}\ ]^T = \mathbf{A}\mathbf{s}\,. \tag{2.6}$$

Equation (2.6) is the one-step shifted version of the graph signal. The k-step shifted version of the signal is denoted as $\widetilde{\mathbf{s}}_k = \mathbf{A}^k\mathbf{s}$. This operation replaces the sample s_n related to the node v_n, with the weighted linear combinations of the signal samples at its neighbors, so

$$\widetilde{\mathbf{s}} = \sum_{m \in \mathcal{N}_n} \mathbf{A}_{n,m}\mathbf{s}_m \, . \tag{2.7}$$

In order to understand the effects of the graph shift operator, we construct a subgraph as it is explained in section 2.1, such as $G' = (\mathcal{V}', \mathbf{A})$, where $\mathcal{V}' = \{v'_0, ..., v'_{N-1}\}$ is a set of $N = 40$ nodes. That nodes correspond to sensors placed at 4 separated places: Florida, Ohio, New Mexico and Oregon. This produces a temperature snapshot with high temperature contrasts. The adjacency matrix $\mathbf{A}$ has been built considering $K = 5$ neighbors.

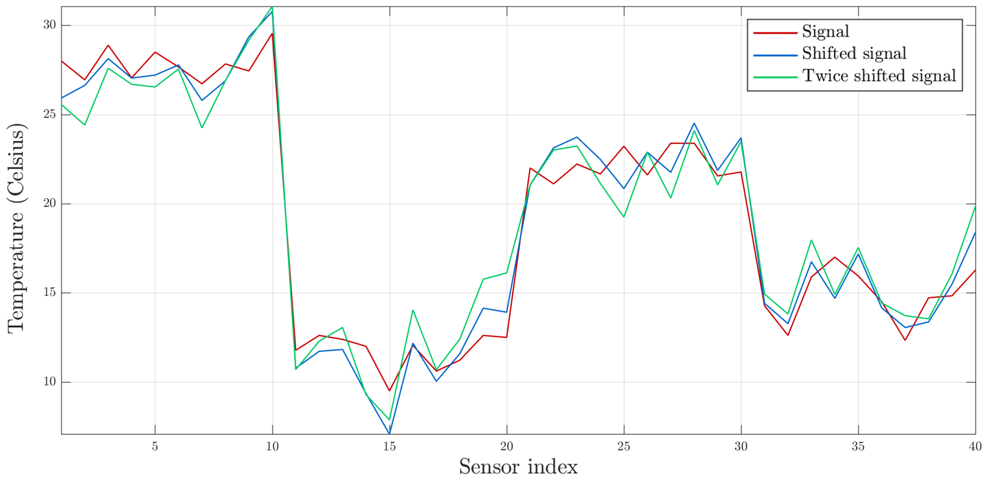

Figure 2.2: Graph shift applied to a snapshot of temperature measurements.

Figure 2.2 shows the effects of applying the graph shift to a signal, and we can extract several conclusions. To guarantee the numerical stability of the shifted signal, we can define the normalized adjacency matrix

$$\bar{\mathbf{A}} = \frac{1}{|\lambda_{max,\mathbf{A}}|}\mathbf{A} \, , \tag{2.8}$$

where $\lambda_{max,\mathbf{A}}$ denotes the maximum eigenvalue of $\mathbf{A}$. That is, $|\lambda_{max,\mathbf{A}}| \geqslant |\lambda_{m,\mathbf{A}}|$ for all $0 < m \leqslant M - 1$, where M is the rank of $\mathbf{A}$ [7]. This guarantees that the shifted signal is not scaled up, as the normalized matrix has spectral norm $||\bar{\mathbf{A}}||_2 = |\lambda_{max,\bar{\mathbf{A}}}| = 1$.

Again, as illustrated in figure 2.2, the shifted signal $\bar{\mathbf{A}}\mathbf{s}$ is an approximation of $\mathbf{s}$. This interesting property is exploited to construct a signal compressor based on graphs, in section 3.1. Another case to study is the effect of graph shifting in terms of signal variance.

2.2.1 Performance of the graph signal variance

Graph signal variance plays an important role in some DSP_G applications such as signal compression on linear prediction, which will be reviewed in chapter 3. In order to achieve better compression ratios, low variance signals are required.

In the following lines we show the procedure to compute the variance of the graph signal $\mathbf{s}$, and can be extrapolated to the case of the shifted graph signal $\widetilde{\mathbf{s}}$.

It is defined the error vector $\boldsymbol{\varepsilon}$ as the difference between the graph signal $\mathbf{s}$ and its mean vector,

$$\boldsymbol{\varepsilon} = \mathbf{s} - \mathbf{m}_s = \mathbf{P}^{\perp}\mathbf{s}\,. \tag{2.9}$$

The mean vector is described as $\mathbf{m}_s = m_s\mathbf{1}$, where m_s is the mean value of the graph signal $\mathbf{s}$, that is $m_s = \frac{1}{N}\mathbf{1}^T\mathbf{s}$. This difference can be defined in terms of graph signal $\mathbf{s}$. Hence, we can define the error vector as $\boldsymbol{\varepsilon} = \mathbf{P}^{\perp}\mathbf{s}$, where $\mathbf{P}^{\perp} = \mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T$ is the orthogonal projector of the mean of the graph signal.

The variance is computed as the squared norm of the error vector defined in (2.9) normalized by the power of the graph signal, that is

$$\text{var}(\mathbf{s}) = \frac{||\boldsymbol{\varepsilon}||^2}{||\mathbf{s}||^2} = \frac{\mathbf{s}^H\mathbf{P}^{\perp}\mathbf{s}}{\mathbf{s}^H\mathbf{s}}\,. \tag{2.10}$$

In equation (2.10) we have taken into consideration the idempotence property of a projector to simplify the expression, that is $\mathbf{P}^{\perp}\mathbf{P}^{\perp} = \mathbf{P}^{\perp}$.

Taking this procedure, we can study the variance of a graph signal. Using the large data set of temperature measurements, we develop two different scenarios. Hence, we build two graphs. The graph $G_1 = (V, \mathbf{A})$ indexes the nearby sensor network with $N = 200$ nodes. The adjacency matrix $\mathbf{A}$ is built as (2.3). The other graph, $G_2 = (V, \mathbf{B})$ indexes the same set of nodes, but the adjacency matrix $\mathbf{B}$ is built as

$$b_{m,n} = \begin{cases} 1 & \text{if } m \in \mathcal{N}_n \\ 0 & \text{otherwise} \end{cases}\,, \tag{2.11}$$

denoting the definition of the edges matrix from the adjacency matrix decomposition described in (2.4).

Figure 2.3 shows the behavior of the variance in terms of the cardinality of the neighborhood. The variance shifted signal $\widetilde{\mathbf{s}_1} = \mathbf{A}\mathbf{s}$ decreases exponentially when largest is the neighborhood of a node, reaching an asymptotic value. This occurs due to the adjacency matrix $\mathbf{A}$ used to create the weighted edges of the graph. Once reached a certain number of neighbors, the network does not make more connections, so the weight of the extra-neighbors edges are 0.

The variance of the shifted signal $\widetilde{\mathbf{s}_2} = \mathbf{Bs}$ has a most unpredictable behavior. We can appreciate two increments in the variance of the shifted signal, caused by relating separated and uncorrelated nodes. In this case, when reaching the full-neighborhood (full connections) the variance drops off.

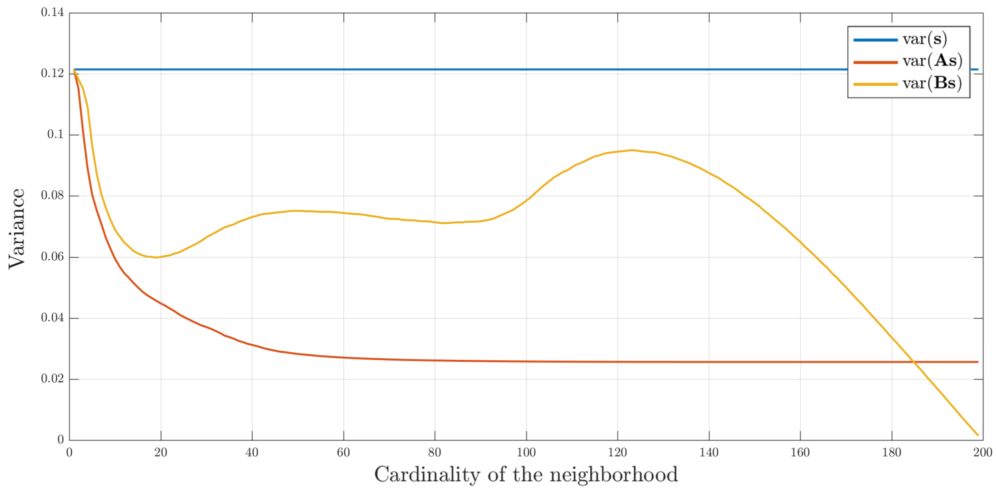

Figure 2.3: Performance of the variance in a nearby sensor network graph.

Figure 2.4 shows the behavior of the variance for the scattered graph G_2. Both profiles are exactly equal to the nearby network scenario, but of a soft spike in the case of $K = 2$ neighbors. Nodes are highly separated, so the values between them have a low correlation. Increasing the cardinality of the neighborhood avoids the noise effects introduced by low-correlated neighbors.

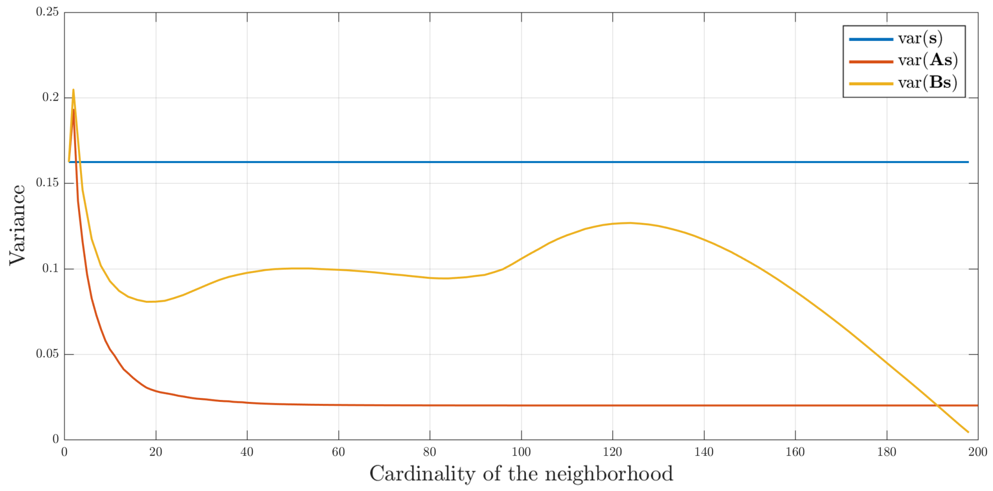

Figure 2.4: Performance of the variance in a scattered sensor network graph.

This experiment demonstrates that shifting a graph signal with an appropriate adjacency matrix reduces its variance, while keeping the resulting signal quite similar to the original one.

2.3 Graph filters

Similarly to the operator introduced in section 2.2, we can represent filtering on a graph using matrix-vector product [2]. Indeed, shifting a signal is a particular case of graph filtering. That is, in general, for an input signal $\mathbf{s}$ filtered by any matrix $\mathbf{H} \in \mathbb{C}^{N \times N}$ yields the filtered graph signal, so

$$\widetilde{\mathbf{s}} = \mathbf{H}\mathbf{s}\,. \tag{2.12}$$

Graph filters are linear, that is the output of the linear combination of these systems is the linear combination of their outputs, so

$$(\alpha \mathbf{H}_1 + \beta \mathbf{H}_2)\mathbf{s} = \alpha \mathbf{H}_1 \mathbf{s} + \beta \mathbf{H}_2 \mathbf{s}\,, \tag{2.13}$$

where $\mathbf{H}_1, \mathbf{H}_2 \in \mathbb{C}^{N \times N}$ are graph filters. Furthermore, another property fulfilled by graph filters is *shift-invariance.* Applying the graph filter to the shifted input signal is equivalent to employ the graph shift to the filtered input signal, so

$$\mathbf{H}(\mathbf{A}\mathbf{s}) = \mathbf{A}(\mathbf{H}\mathbf{s})\,. \tag{2.14}$$

A graph filter $\mathbf{H}$ is linear (2.13) and shift-invariant (2.14) if and only if $\mathbf{H}$ is a polynomial in the graph shift $\mathbf{A}$ [4], that is

$$\mathbf{H} = h(\mathbf{A}) = \sum_{\ell=0}^{L-1} h_\ell \mathbf{A}^\ell = h_0 \mathbf{I} + h_1 \mathbf{A} + \cdots + h_{L-1} \mathbf{A}^{L-1}\,, \tag{2.15}$$

where $h_\ell \in \mathbb{C}$ are the complex filter coefficients, called graph filter *taps.* From (2.15), there exist a limited number of taps [5]. Any graph filter has a unique equivalent filter on the same graph with at most $L = M_\mathbf{A}$, where $M_\mathbf{A}$ is the degree of the minimal polynomial [1] of the adjacency matrix $\mathbf{A}$, assuming that its characteristic and minimal polynomials are equal, $p_\mathbf{A}(x) = m_\mathbf{A}(x)$ [4].

The inverse of a graph filter, if it exists (the graph filter matrix $h(\mathbf{A})$ is non-singular), is also a graph filter $g(\mathbf{A}) = h(\mathbf{A})^{-1}$ on the same graph $G = (V, \mathbf{A})$.

[1] The minimal polynomial of $\mathbf{A}$ is the unique monic polynomial of the smallest degree that annihilates $\mathbf{A}$, that is $m_\mathbf{A}(\mathbf{A}) = 0$.

2.4 Graph Fourier transform

The Fourier transform is plenty used in discrete signal processing. DSP_G defines the Fourier transform with graph filters. Mathematically, a Fourier transform with respect to a set of operators is the expansion of a signal into a basis of the operator's eigenfunctions.

2.4.1 Spectral decomposition

For a graph signal $\mathbf{s}$ from a signal subspace $\mathcal{S}$, $\mathbf{s} \in \mathcal{S}$, the output $\widetilde{\mathbf{s}} = \mathbf{H}\mathbf{s} = h(\mathbf{A})\mathbf{s}$ of any graph filter $h(\mathbf{A})$ (accomplishing the properties defined in section 2.3) also belong to the same signal subspace $\mathcal{S}$.

For the sake of simplicity of the discussion, assume that $\mathbf{A}$ is diagonalizable [2]. Hence, the spectral decomposition (SVD) of the adjacency matrix is

$$\mathbf{A} = \mathbf{V}\boldsymbol{\Sigma}\mathbf{Q}^H \,, \tag{2.16}$$

where $\mathbf{V} = [\ \mathbf{v}_0 \ \ \dots \ \ \mathbf{v}_{R-1}\ ]$ and $\mathbf{Q} = [\ \mathbf{q}_0 \ \ \dots \ \ \mathbf{q}_{R-1}\ ]$ are unitary by the left $P \times R$ and $Q \times R$ matrices respectively. The columns of $\mathbf{V}$ and the columns of $\mathbf{Q}$ are called the left-singular vectors and right-singular vectors of $\mathbf{A}$, respectively. The matrix $\boldsymbol{\Sigma} = diag\,(\,[\ \sigma_0 \ \ \dots \ \ \sigma_{R-1}\ ]^T\,)$ is a diagonal $R \times R$ matrix that contains the singular values. Since this decomposition is too generic, and $\mathbf{A}$ is a square matrix, we can consider the Eigenvalue Decomposition (EVD), that is

$$\mathbf{A} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^H \,, \tag{2.17}$$

where the matrix $\mathbf{V}$ is a square matrix which contains the eigenvectors $\mathbf{v}_k$ of $\mathbf{A}$ and $\boldsymbol{\Lambda}$ is the diagonal matrix that contains its eigenvalues, λ_k.

From (2.17) we can rewrite the expression of the graph filter as

$$\mathbf{H} = h(\mathbf{A}) = \sum_{\ell=0}^{L-1} h_\ell \mathbf{A}^\ell = \sum_{\ell=0}^{L-1} h_\ell (\mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^H)^\ell = \tag{2.18}$$

$$\sum_{\ell=0}^{L-1} h_\ell \mathbf{V}(\boldsymbol{\Lambda})^\ell \mathbf{V}^H = \mathbf{V}(\sum_{\ell=0}^{L-1} h_\ell \boldsymbol{\Lambda}^\ell)\mathbf{V}^H = \mathbf{V}h(\boldsymbol{\Lambda})\mathbf{V}^H \,.$$

In equation (2.18) it has been applied the functional property for an eigenvalue decomposition, so $f(\mathbf{A}) = f(\mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^H) = \mathbf{V}f(\boldsymbol{\Lambda})\mathbf{V}^H$.

[2] In order to simplify the problem, we assume that $\mathbf{A}$ is diagonalizable. However, we generally cannot assume it, which leads us to the Jordan decomposition of $\mathbf{A}$.

2.4.2 Graph Fourier transform

The spectral decomposition of the signal subspace $\mathcal{S}$ expands each signal $\mathbf{s} \in \mathcal{S}$ on the basis defined by the generalized eigenvectors, $\mathbf{s} = \mathbf{V}\,\hat{\mathbf{s}}$. Hence, the vector of expansion coefficients is given by

$$\hat{\mathbf{s}} = [\ \hat{s}_0 \quad \hat{s}_1 \quad \dots \quad \hat{s}_{N-1}\ ]^T = \mathbf{V}^{-1}\mathbf{s}\,. \tag{2.19}$$

We call the expression in (2.19) the *graph Fourier transform* and the coefficients of $\hat{\mathbf{s}}$ the *spectrum* of the graph signal $\mathbf{s}$ [7]-[8]. We can write the *graph Fourier matrix* as

$$\mathbf{F} = \mathbf{V}^{-1}\,. \tag{2.20}$$

The inverse of the graph Fourier transform reconstructs the input signal $\mathbf{s}$ from its spectrum. Moreover, the adjacency matrix – computed as (2.3) – is symmetric, hence is diagonalizable. Then, the graph Fourier matrix is orthogonal, that is $\mathbf{F}^{-1} = \mathbf{F}^H$.

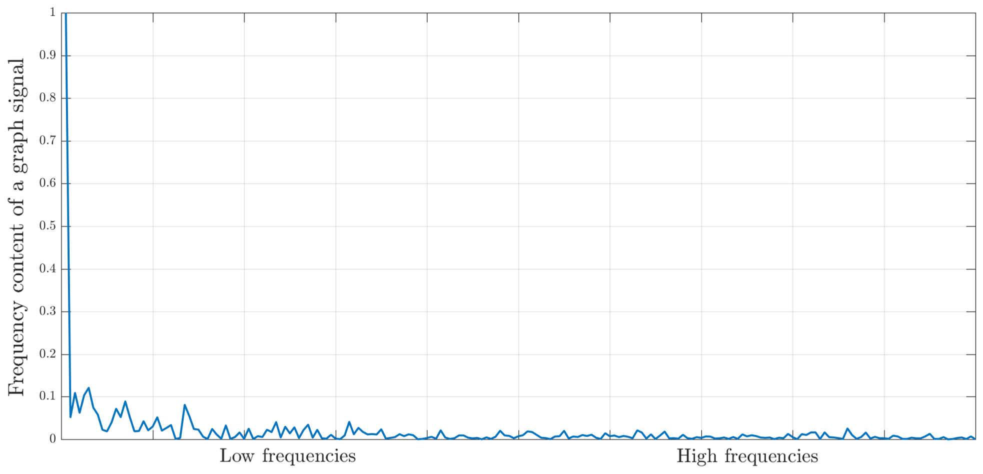

Figure 2.5: Frequency content of a graph signal.

Figure 2.5 shows the plot of the spectrum $\hat{\mathbf{s}}$ of a graph signal. From the temperatures measurements we construct the graph $G = (V, \mathbf{A})$ with $N = 200$ nodes from the state of Colorado. Note that low frequencies have higher values, so the main information of the graph signal is located at these frequencies. This consideration will be useful for the data compression application developed in section 3.1.

3 | Applications on Graphs

In this chapter we are going to develop two important data processing applications. The first one consists on a signal compressor and the second one is a linear predictor. These examples illustrate the effectiveness of DSP_G in front of standard DSP.

3.1 Signal compression

Signal compression is an important tool for reducing both communication costs and storage sizes. These techniques need efficient signal representation. For instance, JPEG expands signals into cosine bases applying the discrete cosine transform (DCT) . Hence, it is expected that most information about the signal is captured with few basis functions.

This expansion of the signal can correspond to a Fourier basis. As we have seen in section 2.4, lower frequencies dominate the spectrum of the graph signal. Hence, we say that a graph signal is sparse in the frequency domain. This important property makes Fourier bases useful for efficient graph signal representation, and therefore for compression [3]-[7]-[8].

3.1.1 Compression algorithm

In section 2.2 we have seen the effects of shifting a graph signal. Given a graph $G = (V, \mathbf{A})$, assuming that the adjacency matrix $\mathbf{A}$ is computed as (2.3), the shifted signal $\widetilde{\mathbf{s}} = \bar{\mathbf{A}}\mathbf{s}$ – where $\bar{\mathbf{A}}$ is the normalized adjacency matrix (2.8) – is a close approximation of input signal $\mathbf{s}$. Hence, we can rewrite shifted signal as

$$\widetilde{\mathbf{s}} = [\ \widetilde{s}_0 \quad \widetilde{s}_1 \quad \dots \quad \widetilde{s}_{N-1}\ ]^T = \mathbf{A}\mathbf{s} = \boldsymbol{\rho} \odot \mathbf{s}\,, \tag{3.1}$$

where $\boldsymbol{\rho} = [\ \rho_0 \quad \rho_1 \quad \dots \quad \rho_{N-1}\ ]^T$ is the vector that scales up input graph signal $\mathbf{s}$. As the shifted signal is a close approximation of $\mathbf{s}$, we can rewrite (3.1) as

$$\widetilde{\mathbf{s}} = \mathbf{A}\mathbf{s} = \boldsymbol{\rho} \odot \mathbf{s} \approx \rho\mathbf{s}\,. \tag{3.2}$$

As shown in figure 2.2, $\rho \approx 1$. This consideration demonstrates that graph signal $\mathbf{s}$ can be expressed as the combination of few eigenvectors of $\mathbf{A}$.

Consider the large dataset of temperature measurements. We can construct the graph $G = (V, \mathbf{A})$ with $N = 2000$ nodes indexed to sensors located around the American territory and the adjacency matrix is computed as (2.8). We compute the Fourier transform matrix $\mathbf{F}$ (2.20) from the eigenvector matrix of $\mathbf{A}$. From each daily snapshot $\mathbf{s}_m$ with $0 < m \leqslant 365$, we compress each signal by keeping only C coefficients of the spectrum signal $\hat{\mathbf{s}}_m = \mathbf{F}\,\mathbf{s}_m$,

$$\check{\mathbf{s}} = \mathbf{F}^{-1}\hat{\mathbf{s}}_{(C)} = \mathbf{F}^{-1}[\begin{array}{cccccccc} \hat{s}_0 & \hat{s}_1 & \dots & \hat{s}_{C-1} & 0 & \dots & 0 \end{array}]^T, \tag{3.3}$$

where $\hat{\mathbf{s}}_{(C)}$ is the spectrum signal $\hat{\mathbf{s}}$ keeping only C coefficients. In equation (3.3) it is assumed that $|\hat{s}_0| \geqslant |\hat{s}_1| \geqslant \ldots \geqslant |\hat{s}_{N-1}|$.

We repeat this procedure for all $M = 365$ daily snapshots $\mathbf{s}_m$. Then, we calculate the average approximation error vector $\boldsymbol{\phi}$ as

$$\phi\,(C) = \frac{\sum\limits_{m=0}^{M-1} ||\check{\mathbf{s}}_{m,C} - \mathbf{s}_m||^2}{\sum\limits_{m=0}^{M-1} ||\mathbf{s}_m||^2}\,. \tag{3.4}$$

In order to demonstrate the validity of signal compression using the Graph Fourier matrix, we compress the signal using classical *Discrete Fourier Transform* (*DFT*). A DFT_N signal is defined as

$$\hat{\mathbf{s}} = \mathbf{DFT}_N\mathbf{s}\,, \tag{3.5}$$

where $\mathbf{s}$ is the input graph signal and $\mathbf{DFT}_N$ is the $N \times N$ *DFT* matrix. The $\mathbf{DFT}_N$ matrix is defined as

$$\mathbf{DFT}_N = \frac{1}{\sqrt{N}}\begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & e^{-j2\pi/N} & e^{-j4\pi/N} & \dots & e^{-j2\pi(N-1)/N} \\ 1 & e^{-j4\pi/N} & e^{-j8\pi/N} & \dots & e^{-j4\pi(N-1)/N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & e^{-j2\pi(N-1)/N} & e^{-j4\pi(N-1)/N} & \dots & e^{-j2\pi(N-1)^2/N} \end{bmatrix}. \tag{3.6}$$

So, from the DFT_N signal introduced in (3.5), we compress the signal keeping only C coefficients. Then, we calculate the average approximation error vector $\boldsymbol{\varphi}$ as described in (3.4).

3.1.2 Results discussion

Next figures show the performance of the average approximation error in terms of the number of coefficients used to reconstruct the input signal.

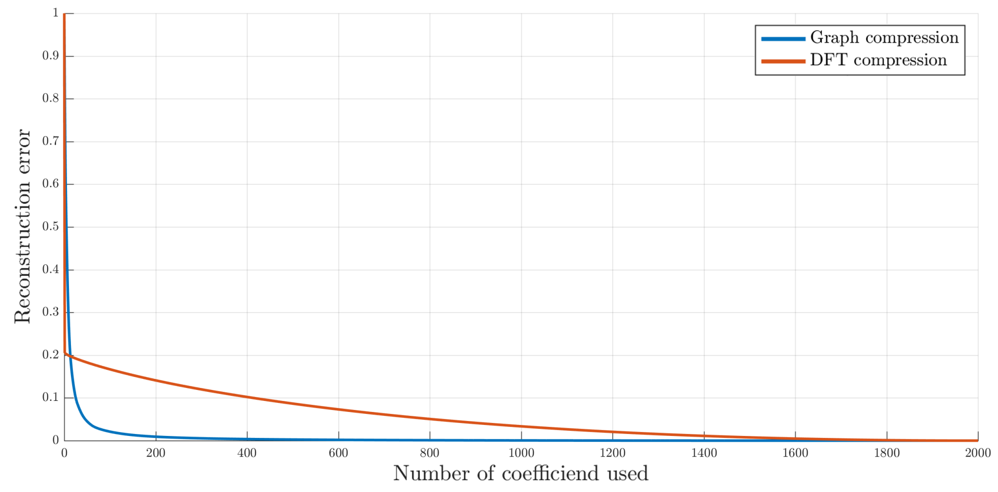

Figure 3.1: Average reconstruction errors ϕ and φ.

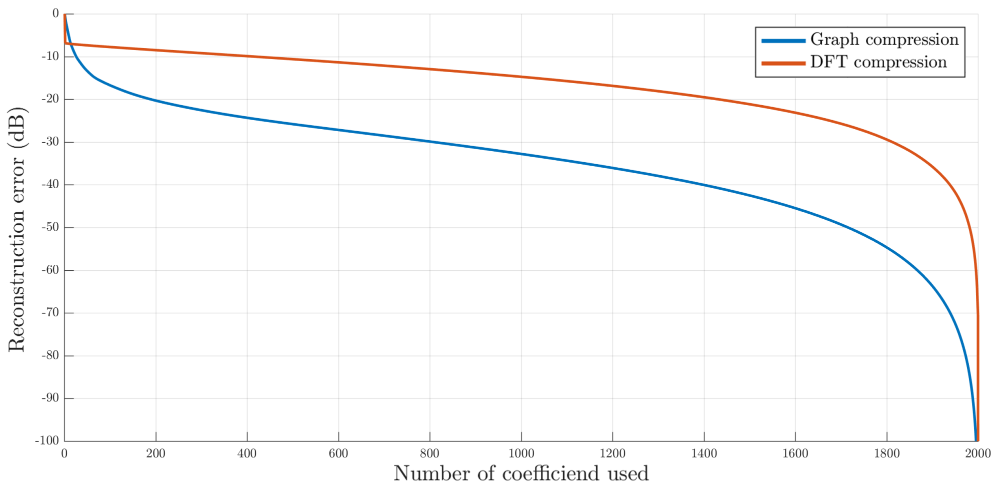

Figure 3.2: Logarithmic average reconstruction errors ϕ and φ.

From both figures, we can see the improvement introduced by graph technique in front of the classical DFT compression. Nevertheless, if we reconstruct the signal taking only one coefficient from the spectrum signal, we get less error with DFT compression technique. Anyway, this tendency only applies in the case of $C = 1$. Note that graph compression error abruptly decreases, taking an insignificant value with only 10% of the coefficients. This can be easily seen in figure 3.2. With $C = 200$ coefficients, graph compression represents a gain of more than 10 dB with respect to DFT compression. In addition, with large number of coefficients used for reconstruction this gain becomes higher.

Figure 3.3 shows the plot of four recovered graph signals $\check{\mathbf{s}}$ for different values of C. Assuming the graph defined with $N = 2000$ nodes, the number of coefficients used for reconstruction are: $C_{(A)} = 10$, $C_{(B)} = 50$, $C_{(C)} = 100$ and $C_{(D)} = 200$.

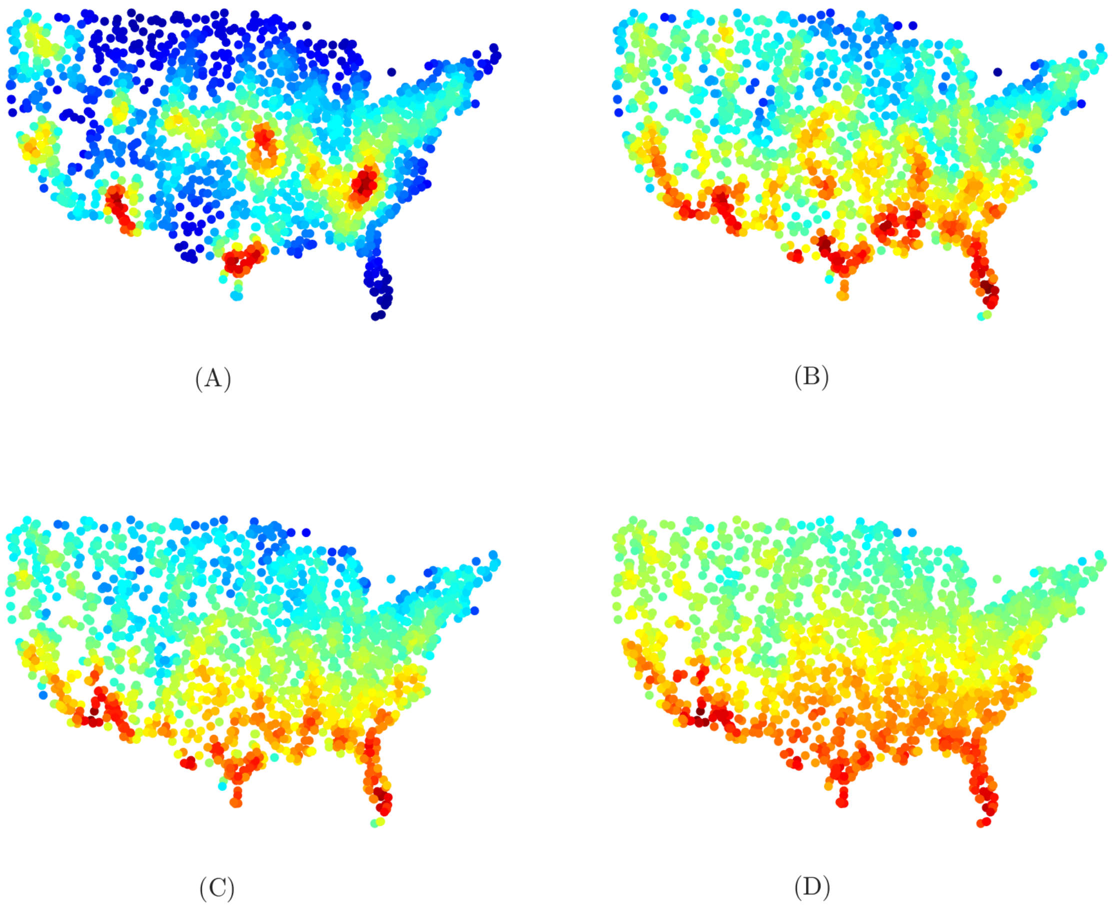

Figure 3.3: Recovered graph signal with different values of C.

In this application we have seen the effects of compressing a graph signal with a novel technique: we have used Graph Fourier transform matrix to expand the signal into the Fourier basis. The compressed signal is obtained by performing the inverse transformation considering only some spectrum coefficients. We have seen that compression ratios of 10% from the original size lead to insignificant errors. This means that the main information of the graph signal is located in a very few number of eigenvectors, representing the lowest frequencies of the original signal. In addition, we have compared the technique to classical DFT compression, getting results significantly better in terms of average reconstruction errors for few coefficients.

3.2 Linear prediction

Linear prediction (*LP*) is a technique used in many applications in discrete signal processing. Mainly, it is applied on highly correlated signals. So, current sample of the signal is approximated (predicted) by a linear combination of the previous ones.

3.2.1 Problem statement

Linear prediction is based on two components: a forward (prediction) filter and a backward (recovery) filter. Figure 3.4 shows the implementation of the forward filter. Backward filter is implemented as the inverse of the forward filter.

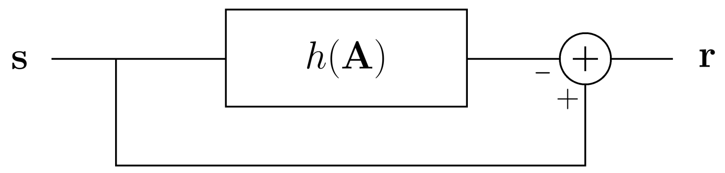

Figure 3.4: Forward filter.

The forward filter permits obtaining a residual signal that can be quantized with few bits. The backward filter recovers an approximation of the original signal from the residual one [3]-[5].

Consider the graph $G = (\mathcal{V}, \mathbf{A})$, where $\mathbf{A}$ is the adjacency matrix built as (2.3) and $\mathcal{V}$ is the set of nodes v_n indexed to $0 < n \leqslant 150$ sensors located all along the American territory. Each one of these sensors is making daily temperature measurements, so we get snapshots $\mathbf{s}_m$ with $0 < m \leqslant 365$, where m is the daily index. For each snapshot is constructed a prediction filter $h(\mathbf{A})$ with L taps. The criterion used to construct the filter is to minimize the energy of the residual output signal $\mathbf{r}$, that is

$$\min_{\mathbf{h}} ||\mathbf{r}||^2 = \min_{\mathbf{h}} ||\mathbf{s} - h(\mathbf{A})\mathbf{s}||^2 \min_{\mathbf{h}} ||(\mathbf{I} - h(\mathbf{A}))\mathbf{s}||^2 \, . \tag{3.7}$$

To avoid the trivial solution $h(\mathbf{A}) = \mathbf{I}$, it has been set $h_0 = 0$, so

$$\begin{pmatrix} h_1 & h_1 & \dots & h_{L-1} \end{pmatrix}^T = (\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\mathbf{s} \, , \tag{3.8}$$

where $\mathbf{B} = \begin{pmatrix} \mathbf{As} & \dots & \mathbf{A}^{L-1}\mathbf{s} \end{pmatrix}$. For further information about the construction of the prediction filter see appendix B.

Figure 3.5 shows the effects of the forward filter illustrated in figure 3.4, and defined as $\mathbf{r} = \mathbf{s} - h(\mathbf{A})\mathbf{s} = (\mathbf{I} - h(\mathbf{A}))\mathbf{s}$.

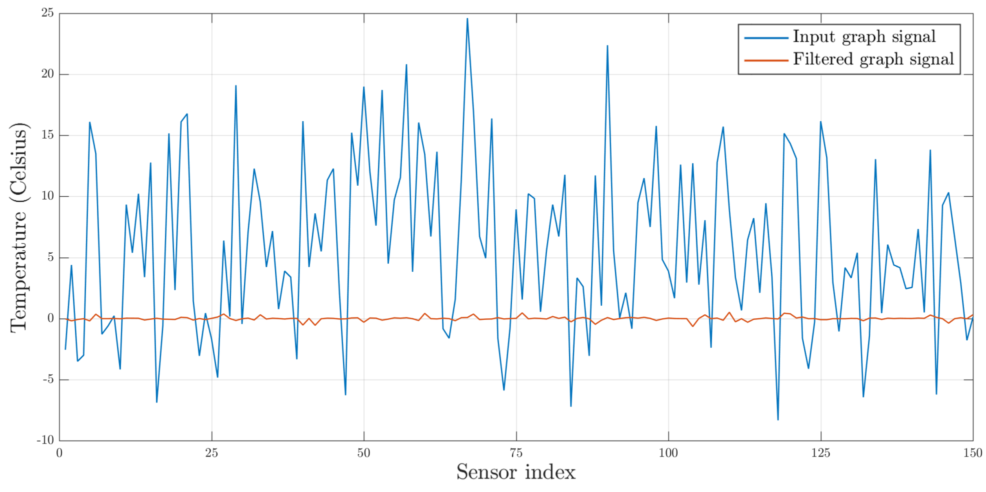

Figure 3.5: Input and output signals from forward filter.

We can appreciate that the residual energy $||\mathbf{r}||^2$ is considerably small compared to the energy of the input graph signal $\mathbf{s}$. The residual signal $\mathbf{r}$ is quantized using B bits. Figure 3.6 shows the quantized signal for nodes v_n with $100 \leqslant n \leqslant 150$ and $B = 2$, that is $2^B = 4$ quantization levels.

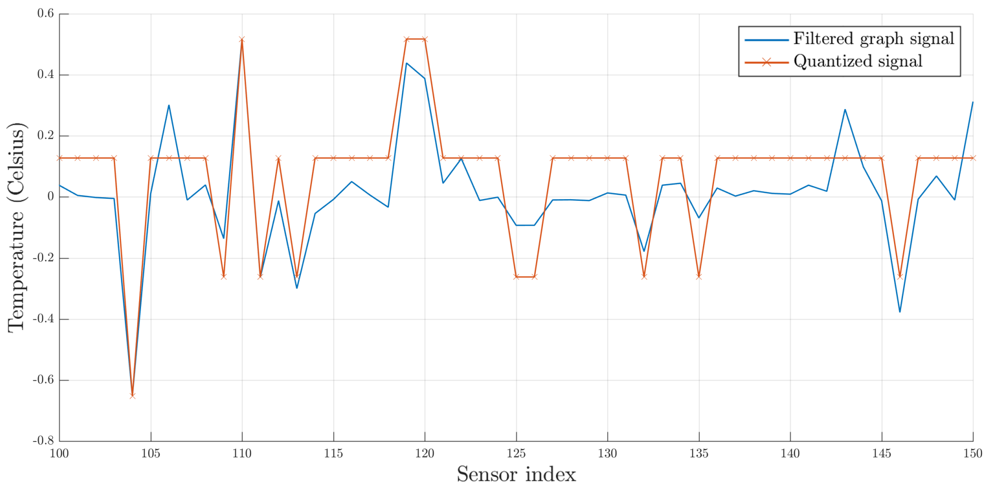

Figure 3.6: Quantization of the residual signal.

Then, the quantized residual signal $\check{\mathbf{r}}$ is processed with the backward filter to obtain an approximation of input graph signal $\mathbf{s}$, that is

$$\check{\mathbf{s}} = (\mathbf{I} - h(\mathbf{A}))^{-1}\check{\mathbf{r}}. \tag{3.9}$$

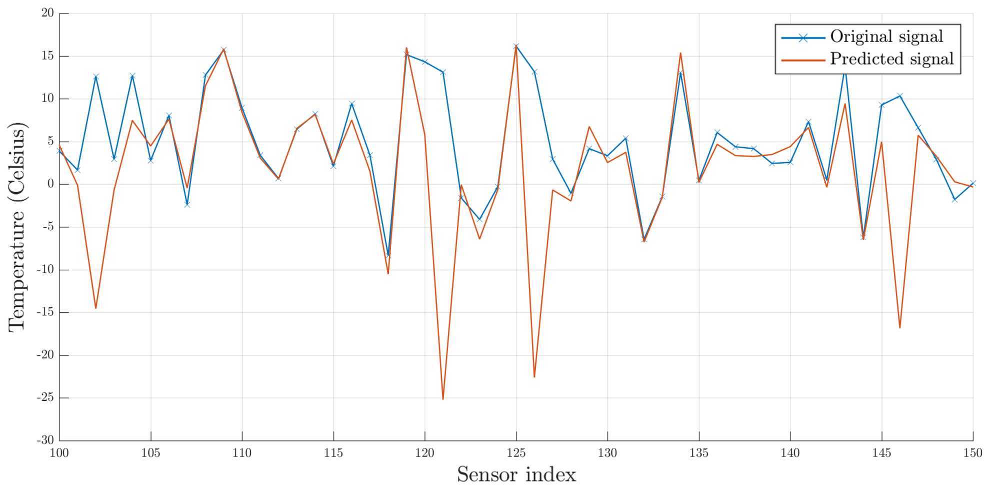

Figure 3.7: Original and approximated graph signal $\mathbf{s}$.

Figure 3.7 shows input and output signals. In order to get better predictions to reduce approximation error, we can change the number of neighbors K considered in the graph, the number of filter taps L used to predict the signal and the number of bits B used to quantize residual signal. Combining these three magnitudes lead us to an optimal solution.

3.2.2 Results discussion

From the graph $G = (V, \mathbf{A})$ with $N = 150$ nodes, we construct subgraphs such as $G'_K = (V, \mathbf{A}_K)$ with $2 \leqslant K \leqslant 15$ neighbors. For each subgraph, we built prediction filters with $2 \leqslant L \leqslant 10$ filter taps. So we got finally 126 different scenarios. Considering the average approximation error as

$$\phi\left(B\right) = \frac{\sum\limits_{m=0}^{M-1} ||\check{\mathbf{s}}_{m,B} - \mathbf{s}_m||^2}{\sum\limits_{m=0}^{M-1} ||\mathbf{s}_m||^2}, \tag{3.10}$$

where $\phi\left(B\right)$ is the average error for each scenario considering a quantization with B bits and M is the total number of snapshots, that is 365 days.

Figures 3.8 and 3.9 show the performance of the average approximation error defined in (3.10) for filters with $L = 3$ and $L = 8$ taps respectively. We can observe that graphs with a large number of neighbors $(9 \leqslant K \leqslant 15)$ lead to lower errors for prediction filters with few number of taps $(2 \leqslant L \leqslant 5)$, whereas graphs with few neighbors $(2 \leqslant K \leqslant 7)$ lead to lower errors for prediction filters with large numbers of taps $(4 \leqslant L \leqslant 8)$. Increasing the number of neighbors and filter taps lead to large errors, due to overfitting graph signals $\mathbf{s}$.

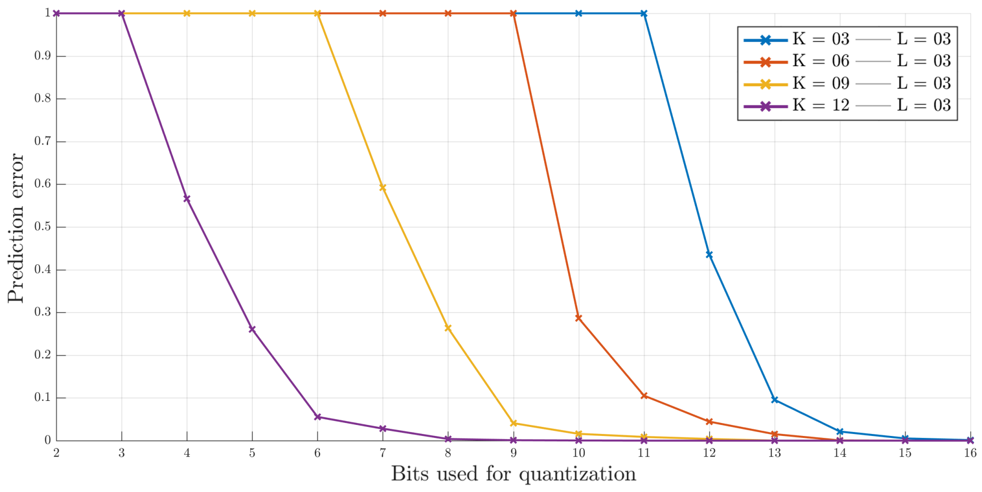

Figure 3.8: Average approximation errors for LP coding with $L = 3$ taps.

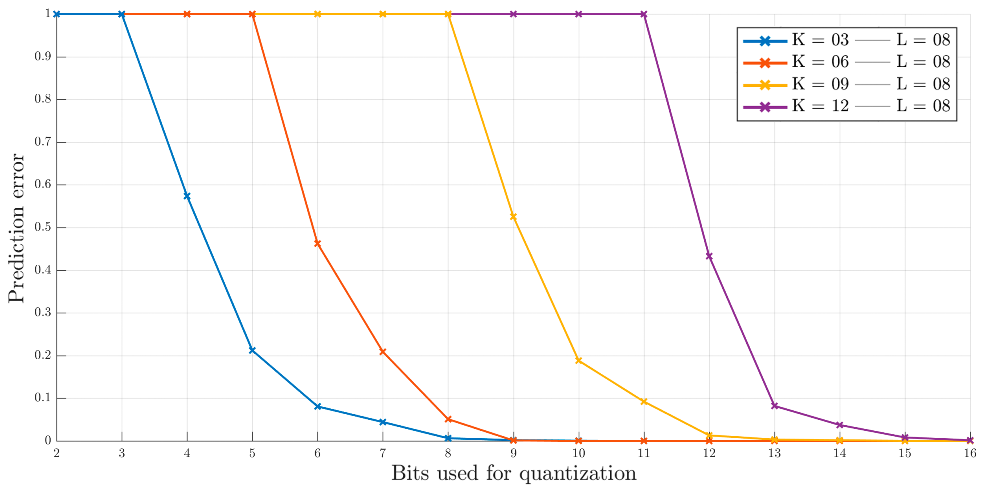

Figure 3.9: Average approximation errors for LP coding with $L = 8$ taps.

In this application we have seen the construction of a linear predictor with graph's techniques. We have exploited the properties of graph filtering such as explained in section 2.3.

We demonstrate that there is a trade-off between graphs and filter parameters. For graphs with large number of neighbors we got better results in terms of average approximation errors with prediction filters whose impulse response has few taps. In contrast to classical linear prediction of DSP, increasing the number of taps lead to large errors for high connected graphs. This concept is a key issue for data filtering on graphs. Constructing filters with large impulse responses require high computational costs.

4 | Adjacency Matrix Identification

Throughout the thesis, we have considered graphs with known structure. So far, we have assumed a popular adjacency model based on distances between nodes, as studied in (2.3). Then, we analyzed how transformations in the structure of the graph impact into the signals. In this chapter we address the problem of identifying the graph structure from the signals on its nodes. This novel idea is accurately studied by S. Segarra et al. in [9]. It is mainly based on considering that a specific graph signal may be seen as the output of a filter when the input is an activation noise. Figure 4.1 shows the concept of identifying the structure of the graph from a random signal.

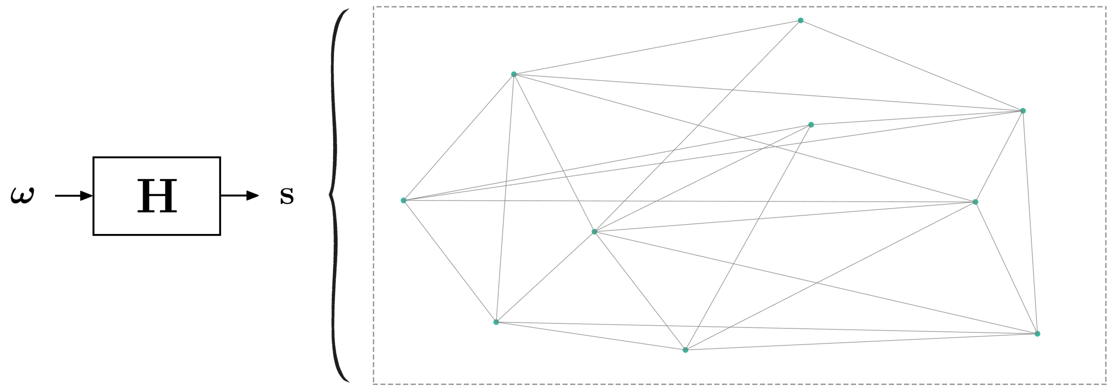

Figure 4.1: Concept schema for the identification of the graph structure.

4.1 Problem statement

The main idea is to identify the structure of a random graph signal. This structure encodes relationships between data related to each node of the graph.

We define the graph signal as $\mathbf{s} = [\ s_0 \quad s_1 \quad \ldots \quad s_{N-1}\ ]^T$, just like in (2.1), where each element $s_n \in \mathbb{R}$ is related to a node of a general graph. Let us suppose that the graph signal is generated from a zero-mean white signal $\boldsymbol{\omega}$ with covariance matrix $\mathbf{C}_\omega = \mathbb{E}\{\boldsymbol{\omega}\boldsymbol{\omega}^H\} = \mathbf{I}$. Hence, we can rewrite $\mathbf{s}$ as

$$\mathbf{s} = \mathbf{H}\boldsymbol{\omega} = h(\mathbf{S})\boldsymbol{\omega} = (\sum_{\ell=0}^{L-1} h_\ell \mathbf{S}^\ell)\boldsymbol{\omega}\,, \tag{4.1}$$

where $h(\mathbf{S})$ is a graph filter obeying the algebra defined in section 2.3 and $\mathbf{S}$ is a general shift matrix. The covariance matrix of the generated graph signal $\mathbf{s}$ is

$$\mathbf{C}_s = \mathbb{E}\{\,(\mathbf{H}\boldsymbol{\omega})(\mathbf{H}\boldsymbol{\omega})^H\,\} = \mathbf{H}\,\mathbb{E}\{\,\boldsymbol{\omega}\boldsymbol{\omega}^H\,\}\,\mathbf{H}^H = \mathbf{H}\mathbf{H}^H\,. \tag{4.2}$$

We can rewrite the graph filter $\mathbf{H} = h(\mathbf{S})$ using the spectral decomposition such as defined in (2.18). Hence, the covariance matrix is defined as

$$\mathbf{C}_s = \mathbf{H}\mathbf{H}^H = \mathbf{V}(\sum_{\ell=0}^{L-1} h_\ell \boldsymbol{\Lambda}^\ell)\mathbf{V}^H\mathbf{V}(\sum_{\ell=0}^{L-1} h_\ell \boldsymbol{\Lambda}^\ell)\mathbf{V}^H = \mathbf{V}\left|\sum_{\ell=0}^{L-1} h_\ell \boldsymbol{\Lambda}^\ell\right|^2 \mathbf{V}^H\,. \tag{4.3}$$

Remember that eigenvector matrix $\mathbf{V}$ is unitary by the left, so $\mathbf{V}^H\mathbf{V} = \mathbf{I}$. Equation (4.3) shows that the only difference between the covariance matrix of the graph signal and the graph shift matrix is on their eigenvalues. Then, we have demonstrated that the matrices $\mathbf{C}_s$ and $\mathbf{S}$ have the same eigenvectors. This concept is the basis on the method developed in this chapter to find the graph structure.

4.1.1 Algorithm criterion and constraints

The scope of this chapter is to consider the graph shift matrix $\mathbf{S}$ as the adjacency matrix $\mathbf{A}$ of the graph. Hence, some considerations have to be taken. One possible criterion to find the structure is on minimizing the total energy of the edges of the graph.

The criterion used in this chapter to find the structure of the graph consists on finding the most efficient matrix, i. e., identifying a sparse adjacency matrix by minimizing the ℓ_1 norm, computed as

$$||\mathbf{S}||_1 = \max_{0\leqslant n\leqslant N-1} \sum_{m=0}^{M-1} |s_{m,n}|\,, \tag{4.4}$$

which is simply the maximum absolute column sum of the matrix. A sparse adjacency matrix denotes a graph with few connections between nodes, that is a reduced neighborhood. Hence, we avoid redundant connections or connections that could introduce some noise.

The graph shift matrix have some properties to accomplish, which are the constraints of the algorithm to identify it. First constraint has been developed before. The eigenvectors of the graph shift matrix are known and are equal to the eigenvectors of the covariance matrix of the graph signal. Hence, we can write a first version of the criterion, so

$$\hat{\mathbf{S}} = \arg\min_{\{\mathbf{S},\boldsymbol{\lambda}\}} ||\mathbf{S}||_1 \qquad \text{s. t} \qquad \mathbf{S} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^H = \sum_{k=0}^{N-1} \lambda_k \mathbf{v}_k \mathbf{v}_k^H\,, \tag{4.5}$$

where $\boldsymbol{\lambda} = \operatorname{diag}(\,\boldsymbol{\Lambda}\,)$ is the vector containing the eigenvalues λ_k from $\mathbf{S}$.

The optimization problem in (4.5) require more constraints in order to consider the graph shift as the adjacency matrix, i. e., prior knowledge about the structure of the graph. These constraints are:

1. Absence of self-loops, that is each diagonal entry of the adjacency matrix must be zero, $s_{ii} = 0 \quad \forall i$.
2. Non-negative weights of the edges, $s_{ij} \geqslant 0 \quad \forall i \neq j$.
3. The adjacency matrix have to be real and symmetric. Hence, it has to be a square matrix.
4. In order to avoid trivial solution $\hat{\mathbf{S}} = \mathbf{0}$, we fix the weights of the first node to 1, $\sum_j s_{j1} = 1$.

4.2 Identifying the adjacency matrix

In order to incorporate all the constraints of the adjacency matrix, we need to make some accommodations in order to apply a well-known sparsity solution (Basis pursuit).

4.2.1 Algorithm accommodation

First requirement is to transform the problem of identifying the matrix into the identification of a vector. This key step is needed in order to take advantage of ℓ_1-norm minimization algorithms. We can rewrite the graph shift matrix as $\mathbf{S} = [\ \mathbf{s}_0 \quad \dots \quad \mathbf{s}_{N-1}\ ]$. Hence, the vectorization of the matrix results

$$\mathbf{s} = \text{vec}(\mathbf{S}) = [\ \mathbf{s}_0^T \quad \mathbf{s}_1^T \quad \dots \quad \mathbf{s}_{N-1}^T\ ]^T , \tag{4.6}$$

where $\mathbf{s}$ is a column vector with N^2 entries . In order to adapt the constraint from (4.5) to apply into the vector $\mathbf{s}$, we define

$$\mathbf{W} = \mathbf{V} \diamond \mathbf{V} , \tag{4.7}$$

where $\mathbf{V}$ is the eigenvector matrix from the covariance – or adjacency – matrix and $\mathbf{W}$ is a $N^2 \times N$ matrix. Note that matrices are related by the Khatri-Rao (column-wise Kronecker) product [11]. Considering the case $N = 2$, the matrix $\mathbf{W}$ results

$$\mathbf{W} = \mathbf{V} \diamond \mathbf{V} = \begin{bmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \end{bmatrix} \diamond \begin{bmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \end{bmatrix} = \begin{bmatrix} v_{11}v_{11} & v_{12}v_{12} \\ v_{11}v_{21} & v_{12}v_{22} \\ v_{21}v_{11} & v_{22}v_{12} \\ v_{21}v_{21} & v_{22}v_{22} \end{bmatrix} . \tag{4.8}$$

Then, we can rewrite the spectral decomposition of the vectorized matrix $\mathbf{s}$ as

$$\mathbf{s} = \mathbf{W}\boldsymbol{\lambda}\,, \tag{4.9}$$

and the modified criterion defined in (4.5) as

$$\hat{\mathbf{s}} = \arg\min_{\{\mathbf{s},\boldsymbol{\lambda}\}} ||\mathbf{s}||_1 \qquad \text{s. t} \qquad \mathbf{s} = \mathbf{W}\boldsymbol{\lambda}\,. \tag{4.10}$$

Some steps still remain. Note that the criterion depends on two variables, $\mathbf{s}$ and $\boldsymbol{\lambda}$. Moreover, the constraints are not considered yet. We can solve the dependency problem assuming $\boldsymbol{\lambda} = \mathbf{W}^{\#}\mathbf{s}$, where $\mathbf{W}^{\#}$ is the Moore-Penrose pseudoinverse matrix of $\mathbf{W}$. Hence, we can rewrite the constraint $\mathbf{s} = \mathbf{W}\boldsymbol{\lambda}$ as

$$\mathbf{s} = \mathbf{W}\boldsymbol{\lambda} = \mathbf{W}(\mathbf{W}^{\#}\mathbf{s}) \quad \Longleftrightarrow \quad (\mathbf{I} - \mathbf{W}\mathbf{W}^{\#})\,\mathbf{s} = \mathbf{M}\,\mathbf{s} = \mathbf{0}\,, \tag{4.11}$$

where $\mathbf{M} = \mathbf{I} - \mathbf{W}\mathbf{W}^{\#}$ is a $N^2 \times N^2$ projection matrix. Let us define the set of indices $\mathcal{D}$ containing the indices of $\mathbf{s}$ corresponding to the diagonal entries from the graph shift matrix $\mathbf{S}$. From $\mathcal{D}$, denote as $\mathcal{D}^c$ the complement of $\mathcal{D}$, containing the indices for the rest of entries. Taking as an example a graph with $N = 3$ nodes, we obtain the sets $\mathcal{D} = \{1, 5, 9\}$ and $\mathcal{D}^c = \{2, 3, 4, 6, 7, 8\}$. Then, from constraint 1, we can write

$$\mathbf{s}_{\mathcal{D}} = \mathbf{0}\,, \tag{4.12}$$

where $\mathbf{s}_{\mathcal{D}}$ is the subset of $\mathbf{s}$ with $(N^2 - N)$ entries corresponding to the diagonal values from the graph shift matrix. From constraint 4 we can rewrite $\sum_j s_{j1} = 1$ as

$$\sum_j s_{j1} = 1 \quad \Longleftrightarrow \quad (\mathbf{e}_1 \otimes \mathbf{1}_N)^T \mathbf{s} = 1\,, \tag{4.13}$$

where $\mathbf{e}_1 = [\ 1 \quad 0 \quad \dots \quad 0\ ]^T$ is the first canonical basis vector, and $\mathbf{1}_N$ is a ones vector with N entries. Note that vectors are related by the Kronecker product [10]-[11]. For two general matrices $\mathbf{X}_{M\times N}$ and $\mathbf{Y}_{P\times Q}$, the Kronecker product between them results the $MP \times NQ$ matrix such as

$$\mathbf{X} \otimes \mathbf{Y} = \begin{bmatrix} x_{11}\mathbf{Y} & \cdots & x_{1N}\mathbf{Y} \\ \vdots & \ddots & \vdots \\ x_{M1}\mathbf{Y} & \cdots & x_{MN}\mathbf{Y} \end{bmatrix}. \tag{4.14}$$

Hence, $(\mathbf{e}_1 \otimes \mathbf{1}_N) = [\ 1 \quad \dots \quad 1 \quad 0 \quad \dots \quad 0\ ]^T$ is a N^2 vector with ones in the first N positions and zeros elsewhere.

We can join both constraints into one. From the set of indiced defined before, the constraint in (4.11) results

$$\mathbf{M}\,\mathbf{s} = \mathbf{M}_{\mathcal{D}}^T\,\mathbf{s}_{\mathcal{D}} + \mathbf{M}_{\mathcal{D}^c}^T\,\mathbf{s}_{\mathcal{D}^c} = \mathbf{0} \quad \Longleftrightarrow \quad \mathbf{M}_{\mathcal{D}^c}^T\,\mathbf{s}_{\mathcal{D}^c} = \mathbf{0}\,, \tag{4.15}$$

where $\mathbf{M}_{\mathcal{D}}^T\,\mathbf{s}_{\mathcal{D}} = \mathbf{0}$ as defined in (4.12) and $\mathbf{M}_{\mathcal{D}^c}$ is a $(N^2-N)\times N^2$ matrix. Then, we construct the matrix

$$\mathbf{K} = [\ \mathbf{M}_{\mathcal{D}^c} \quad (\mathbf{e}_1 \otimes \mathbf{1}_N)_{\mathcal{D}^c}\ ]\,, \tag{4.16}$$

where $(\mathbf{e}_1\otimes\mathbf{1}_N)_{\mathcal{D}^c}$ is a (N^2-N) vector. Equivalently, we can define this vector as $(\mathbf{e}_1\otimes\mathbf{1}_N)_{\mathcal{D}^c} = (\mathbf{e}_1 \otimes \mathbf{1}_{N-1})$. The resulting $\mathbf{K}$ is a $(N^2 - N) \times (N^2 + 1)$ matrix. Finally, we can write the linear system to be solved by the minimization criterion as

$$\mathbf{K}^T\,\mathbf{s}_{\mathcal{D}^c} = \mathbf{b}\,, \tag{4.17}$$

where $\mathbf{b} = [\ 0 \quad \ldots \quad 0 \quad 1\ ]^T$ is a $(N^2 + 1)$ vector with null entries except for the last one. Note that equation (4.17) is the relation of the equivalences in (4.15) and (4.13). As we adapt the criterion including all the constraints, we can write the optimization problem as

$$\hat{\mathbf{s}}_{\mathcal{D}^c} = \arg\min_{\mathbf{s}_{\mathcal{D}^c}} ||\mathbf{s}_{\mathcal{D}^c}||_1 \qquad \text{s. t} \qquad \mathbf{K}^T\,\mathbf{s}_{\mathcal{D}^c} = \mathbf{b}\,. \tag{4.18}$$

We have found the Basis pursuit problem to identify the optimal structure for a graph with the constraints required for an adjacency matrix. We can rebuild matrix $\hat{\mathbf{S}}$ reshaping the vectorized $\hat{\mathbf{s}}$ as an $N \times N$ matrix,

$$\hat{\mathbf{S}} = [\ \hat{\mathbf{s}}_0 \quad \hat{\mathbf{s}}_1 \quad \ldots \quad \hat{\mathbf{s}}_{N-1}\ ]^T\,, \tag{4.19}$$

where $\hat{\mathbf{s}} = \text{vec}(\hat{\mathbf{S}}) = [\ \hat{\mathbf{s}}_0^T \quad \hat{\mathbf{s}}_1^T \quad \ldots \quad \hat{\mathbf{s}}_{N-1}^T\ ]^T$ such as defined in (4.7), and each $\hat{\mathbf{s}}_n$ is a subvector with N entries.

4.2.2 Results discussion

The problem defined in (4.18), which takes into account the constraints, is now affordable and can be solved by ℓ_1-norm algorithms [12] – [13]. However, constraint 2 has not been taken into account and it is necessary to understand the meaning of negative connections.

In this section we are going to evaluate and compare the identified adjacency matrix with the one defined in (2.3). Consider the graph $G = (\mathcal{V}, \hat{\mathbf{A}})$ where $\mathcal{V} = \{v_0, ..., v_{N-1}\}$ is the set of $N = 50$ nodes indexed to weather stations located around the United States territory. The weighted adjacency matrix $\hat{\mathbf{A}}$ is build by solving the problem in (4.18).

In order to evaluate the behavior of the identified adjacency matrix, we set to null all the residual connections. The criterion chosen to denote these residual connections is

$$\hat{a}_{m,n} = \begin{cases} 0 & \text{if } \hat{a}_{m,n} < 0.01\,\lambda_{max} \\ \hat{a}_{m,n} & \text{otherwise} \end{cases} , \tag{4.20}$$

where λ_{max} denotes the maximum eigenvalue of $\hat{\mathbf{A}}$. Figure 4.2-(A) depicts the binarized values of the identified solution $\hat{\mathbf{A}}$. Blank entries indicate no connection, while dark-blue entries indicate connection between the m-th and n-th nodes. We can appreciate that the criterion used to construct the matrix is not considering distances between nodes. If so, big blank areas will appear to avoid connections between far sensors.

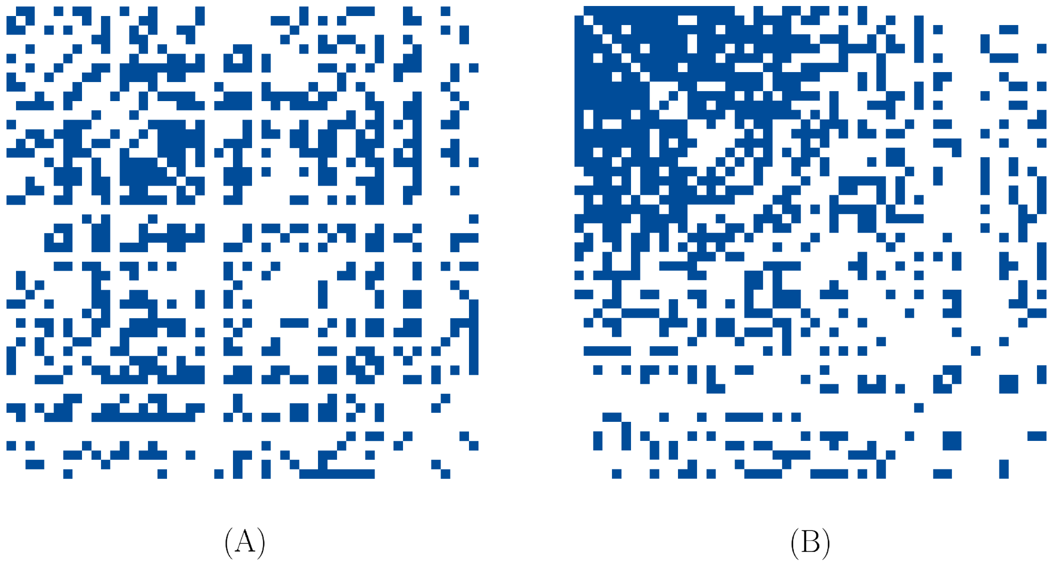

Figure 4.2: Binarized map from the identified adjacency matrix

In order to make a map more representative, we can sort the graph signal from large magnitude values (hot weather) to lowest ones (cold weather). The effects of doing this transformation are shown in 4.2-(B). The graph constructed with this adjacency matrix is connecting large magnitude nodes, and it disconnects the ones with low energy. Even so, it is also connecting high correlated nodes, that is in this example that sensors with similar temperature measurements.

4.2.3 Numerical experiment

In previous section we have seen the behavior of the identified structure, relating nodes with large energy and isolating low-energy ones. The aim of this section is to exploit this property.

Consider the graph $G = (\mathcal{V}, \hat{\mathbf{A}})$ where $\mathcal{V} = \{v_0, ..., v_{N-1}\}$ is the set of $N = 150$ nodes indexed to weather stations located around the United States territory. The weighted adjacency matrix $\hat{\mathbf{A}}$ is build by solving the problem in (4.18). Given the graph, the set of nodes forms the graph signal $\mathbf{s}$. Hence, we can generate shifted signal as

$$\tilde{\mathbf{s}} = \hat{\mathbf{A}}\mathbf{s}\,. \tag{4.21}$$

Shifting by the identified matrix $\hat{\mathbf{A}}$ expands the dynamic range of the signal. In addition, most of the energy of the signal is concentrated in large and low values. That is, it transforms a soft and flat graph signal to an almost binarized one. Then, we define a threshold to binarize the signal such as

$$\xi = \frac{1}{N}\mathbf{1}^T\tilde{\mathbf{s}} \;\approx\; \frac{1}{2}(\tilde{\mathbf{s}}_{\max} - \tilde{\mathbf{s}}_{\min})\,. \tag{4.22}$$

We have defined the threshold as the mean of the graph signal, which is quite similar to the balance point of the signal's energy.

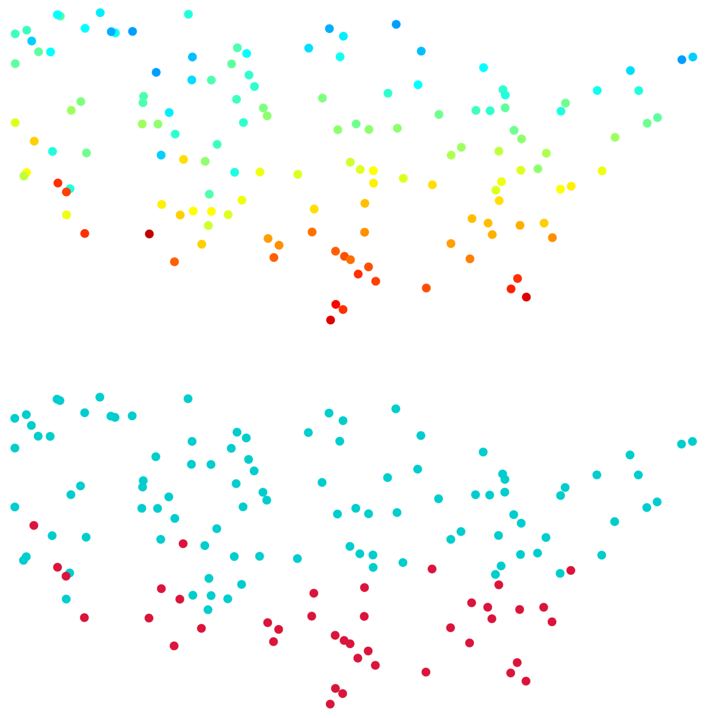
Figure 4.3: Original and binarized graph signals.

Figure 4.3 shows both original and binarized graph signals. Effectively, it corresponds to a temperature map of the United States, where the south is hot and the north is cold.

5 | Conclusions and Future Work

5.1 Conclusions

In this work we have developed a solid background in DSP_G, extending fundamental concepts from classical DSP such as data filtering, spectral decomposition, etc. to large data sets by indexing signals to the nodes of a graph. We have demonstrated that DSP_G is a valid solution to treat amounts of waste data providing some applications. Finally, we faced the problem of identifying the structure of a graph.

In chapter 2 we have reviewed important DSP concepts, building an immersive framework in graph theory. We begin by defining most elemental elements of the graph, such as nodes and structure. We follow by introducing most important operators from classical DSP on graphs, such as shifting, filtering, spectral decomposition and Fourier transform. In all these transformations it is imperative knowing the structure of the graph, hence its adjacency matrix. The procedure used in this work is a popular choice for construction of similarity graphs based on distances between nodes.

In chapter 3 we applied the concepts introduced previously two develop two important applications in DSP. First one consists on signal compression, based on the spectrum signal manage. Fourier transform matrix takes an important role. Comparing the graph compression methodology to a classical technique using DFT matrix, we conclude that we get better compression ratios in relation with the coefficients used to reconstruct original signal. The other application consists on a linear predictor. We tested the behavior of graph filtering in a large range of scenarios as a function of the cardinality of the neighborhood and the coefficients used to build the impulse response of the filter. We realized that increasing the number of coefficients lead to high errors for high-connected graphs, which contradicts with classical linear prediction methodologies.

Finally, we abroad the problem of identifying the structure of a graph. During all the work we have considered a known adjacency matrix build on similarity of the nodes through distances. But this criterion is not unique. We considered the construction of a representative graph by minimizing the connections between nodes. The results obtained gave an other perspective of graph signals, also useful from other applications. Connections are made depending on the node's energy, connecting strongly the highest ones.

5.2 Future work

This degree thesis has built a solid background in DSP on graphs. This novel technique opens a large range of disciplines and research problems.

Following the applications developed in this work, we can develop a robust technique to detect malfunction in sensors networks by band-pass filtering signals. Furthermore, we can extend the DSP_G framework to face problems less common in classical DSP, such as data classification.

On the other hand, we introduce the problem of constructing graphs. Knowing the optimal structure is not obvious and requires a deep learning of the original signal. Lot of literature is developing algorithms and methodologies to face this novel problem, such as the developed in chapter 4. This issue motivates an important problem on DSP_G. We are assisting to the beginning of an important discipline due to the amounts of waste data and massive data sets, the big data problem.

References

[1] "National climatic data center. 1981 - 2010 climate normals", `ftp://ftp.ncdc.noaa.gov/pub/data/`.

[2] Aliaksei Sandryhaila, José M. F. Moura. "Big Data Analysis with Signal Processing on Graphs". *IEEE Signal Processing Magazine*, September 2014.

[3] Aliaksei Sandryhaila, José M. F. Moura. "Discrete Signal Processing on Graphs". *IEEE Trans. Signal Processing*, 2013.

[4] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation", 2003.

[5] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs: Graph filters". *IEEE Trans. Signal Processing*, 2013.

[6] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs," *IEEE Signal Processing Magazine*, 2013.

[7] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs: Frequency Analysis". *IEEE Trans. Signal Processing*, 2014.

[8] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs: Graph Fourier transform". *IEEE Trans. Signal Processing*, 2013.

[9] S. Segarra, A. G. Marques, G. Mateos, and A. Ribeiro, "Network topology identification from spectral templates". *IEEE Statical. Signal Processing Workshop*, 2016.

[10] J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos, and Z. Ghahramani, "Kronecker graphs: an approach to modeling networks". *Journal of Machine Learning Research*, 2010.

[11] S. Liu, G. Trenkler, "Hadamard, Khatri-Rao, Kronecker and other matrix products". *International Journal of Information and Systems Sciences*, 2008.

[12] M. P. Friedlander and M. Saunders, "A Matlab solver for sparse optimization", `https://www.cs.ubc.ca/ mpf/asp-a-matlab-solver-for-sparse-optimization.html`

[13] E. van den Berg and M. P. Friedlander, "Sparse optimization with least-squares constraints". *SIAM J. on Optimization*, 2011.

A | Time planning

Figure A.1 shows the Gantt diagram of the Degree's Thesis. For an accurate explanation of each work-package, we refer to section 1.1.

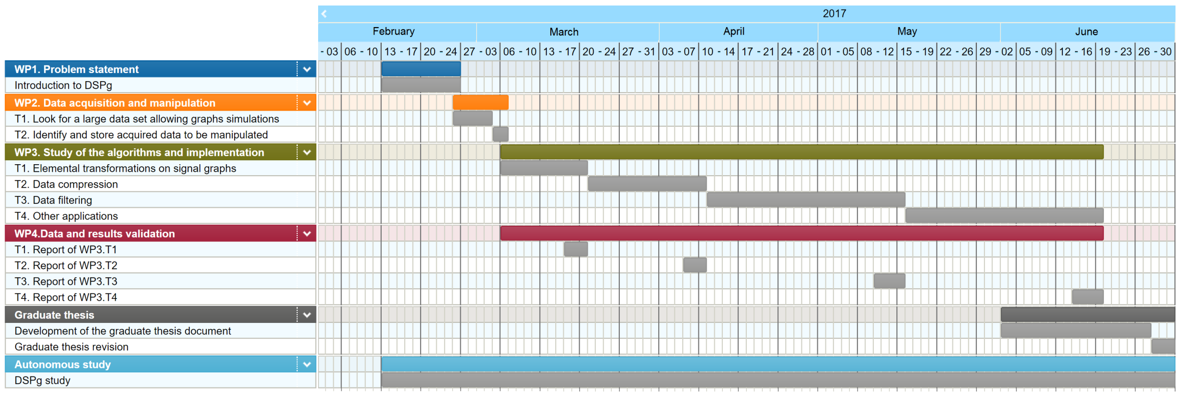

Figure A.1: Time planning of the Degree's Thesis.

Note that during all along the project an autonomous study is needed in order to a suitable development of the work.

B | Prediction filter construction for minimum output energy

Prediction filter is constructed by minimizing the energy of the residual output signal $\mathbf{r}$, that is

$$\min_{\mathbf{h}} ||\mathbf{r}||^2 = \min_{\mathbf{h}} ||(\mathbf{I} - h(\mathbf{A}))\mathbf{s}||^2 \tag{B.1}$$

First, it is appropriate to rewrite the expression $h(\mathbf{A})\mathbf{s}$ because it would be useful in following steps, such as

$$h(\mathbf{A})\mathbf{s} = (h_0\mathbf{I} + h_1\mathbf{A} + h_2\mathbf{A}^2 + \cdots + h_{L-1}\mathbf{A}^{L-1})\mathbf{s} = \mathbf{B}\,\mathbf{h} \tag{B.2}$$

where $\mathbf{B} = \begin{pmatrix} \mathbf{A}^0\mathbf{s} & \mathbf{A}^1\mathbf{s} & \dots & \mathbf{A}^{L-1}\mathbf{s} \end{pmatrix}$ and $\mathbf{h} = \begin{pmatrix} h_0 & h_1 & \dots & h_{L-1} \end{pmatrix}^T$.

In order to solve the expression in (B.1), it is recommended to develop the expression of the residual output signal energy $||\mathbf{r}||^2$.

$$||\mathbf{r}||^2 = ||(\mathbf{I} - h(\mathbf{A}))\mathbf{s}||^2 = ((\mathbf{I} - h(\mathbf{A}))\mathbf{s})^H((\mathbf{I} - h(\mathbf{A}))\mathbf{s}) = \tag{B.3}$$

$$= \mathbf{s}^H\mathbf{s} - \mathbf{s}^H h(\mathbf{A})\mathbf{s} - \mathbf{s}^H(h(\mathbf{A}))^H\mathbf{s} + \mathbf{s}^H(h(\mathbf{A}))^H h(\mathbf{A})\mathbf{s} =$$

$$= \mathbf{s}^H\mathbf{s} - \mathbf{s}^H\mathbf{B}\mathbf{h} - \mathbf{h}^H\mathbf{B}^H\mathbf{s} + \mathbf{h}^H\mathbf{B}^H\mathbf{B}\mathbf{h}$$

In this point, expression (B.1) is easily solvable as follows.

$$\min_{\mathbf{h}} ||\mathbf{r}||^2 = \min_{\mathbf{h}} ||(\mathbf{I} - h(\mathbf{A}))\mathbf{s}||^2 = \tag{B.4}$$

$$= \nabla_{\mathbf{h}^H} ||\mathbf{r}||^2 = \mathbf{B}^H\mathbf{B}\mathbf{h} - \mathbf{B}^H\mathbf{s} = 0$$

By few operations, the graph filter taps vectors results

$$\mathbf{h} = (\mathbf{B}^H\mathbf{B})^{-1}\mathbf{B}^H\mathbf{s} \tag{B.5}$$